# Machine Learning

## #What is machine learning?

Machine learning is a set of techniques that are being developed to identify patterns in data for situations where we can’t use conventional mathematical models.

**A problem will be categorised as a Machine Learning problem if –**

1. A lot of data is present
2. There exists a pattern in our data
3. We cannot pin down a mathematical equation for it.

**Examples in our daily life** – Google search engine, filter out unwanted email, face and fingerprint recognition, suitable match on dating apps.

**Examples for Actuarial Work** – Risk Classification, identifying fraudulent claims and tax declarations

**Under mathematical model**, we use the SAME DATASET for modelling and testing, whereas under machine learning, the dataset is divided into groups for training, validation and testing.

### Machine learning disadvantage

* In statistics, ‘fit’ refers to how well the target function is approximated.
* *Over fitting* refers to learning the training data detail and noise which leads to poor generalisation. It can be limited by using resampling and defining a validation dataset. Here, the algorithm looks for a pattern that does not exist in real life
* *Under fitting* refers to poor inductive learning from training and poor generalisation.

A diagram of different types of data

Description automatically generated

### Feature Scaling

Clustering algorithms require a measure of the distance between each observation on each feature. If the data are not scaled, the weight given to a feature by the algorithm will depend on the units of measurement, as the absolute distances between the observations will vary more for some features than others.

E.g. if one variable varies from 0.03 to 219.66 and the other varies 0.007 to 0.77 only, we need to scale the data.

### Standardisation

, values end up with zero mean and unit standard deviation.

### Normalization

, values end up between 0 and 1.

## #What is Feature Engineering?

* Feature engineering is the process of extracting and organizing the important features from raw data in such a way that it fits the purpose of the machine learning model. It can be thought of as the art of selecting the important features and transforming them into refined and meaningful features that suit the needs of the model.
* Involves selecting relevant features, handling missing values, encoding data and normalizing it.

### Importance?

* Raw data collected via open sources like internet, surveys and reviews contain missing values, unstructured data, incorrect inputs and outliers.
* If we directly use this raw, unprocessed data to train our models, we will land up with a model having a very poor efficiency.

### Benefits?

* Higher model efficiency
* Easier algorithms that fit the data
* Easier for algorithm to detect patterns in the data
* Greater flexibility of the features

## #Handling Missing Data

* **Delete Entire Column Of Feature** which contains a lot of missing values and may not contribute to prediction of output. We may set a threshold value say 70%, so if missing values are more than 70% we remove the entire column
  + Benefit – dimensionality reduction + less complex to compute
  + Drawback – loss of information
* **Impute Missing Values For Continuous Variable** – replace missing values with mean, median or mode.
* **Impute Missing Values For Categorical Variable** – remove data entry (entire row), assign ‘U’ (unknown) or any other dummy variable to missing values. replace missing value with the highest frequency value.
* **Predict Missing Values –** treat the column with missing values as the dependent variable and the rest of the column as independent variables. Now we can using the linear regression or classification models to predict the values. since this method takes into account the correlation between the missing value column and other columns to predict the missing values, it yields much better results than the previous methods.

## #Encoding Independent Variable

Variable country has values : India, Spain and Belgium

* Should we encode it as 0, 1 and 2?

No! because it implies that there is some sort of sequential relationship between the countries.

* Instead → create 3 columns each for India, Spain and Belgium with values 0 and 1.

This can be done with the help with One Hot Encoding.

## #Types of Machine Learning

1. **Supervised Learning** – if the computer is told what the target of the analysis is, then the problem will be categorised as supervised machine learning.

E.g. if we want to predict claim amount or mortality rate then these are ***specific targets***. The outcome could be numerical or categorical. If the outcome is numerical, the problem is categorised as a Regression Analysis. If the outcome is categorical, the problem is categorised as classification problem.

E.g. Generalised linear models, Naive Bayes classification, decision trees, prediction of future lifetime, neural networks, prediction of claims on certain classes of insurance, defaulting of loan, regression models, logistic regression, Probit models, discriminant analysis, perceptron, support vector machines.

1. **Unsupervised Learning** – if the algorithm is not trying to predict an outcome instead it is trying to create a set of categories. Then the problem is categorised as unsupervised learning. The idea is that the data in any one category should be homogeneous as possible and each category should be as different as possible.

E.g. K Means Clustering, Principal components analysis, Apriori algorithm, market basket analysis, text analysis, neural networks

Examples of unsupervised learning techniques include cluster analysis, and the use of association rules such as the apriori algorithm.

The **apriori algorithm** is a machine learning technique that identifies combinations of data values that frequently occur together in a data set, E.g. where users of a music website will tend to download items by the same artist or items of the same genre. It can be used by online retailers as the basis for the ‘Other customers also bought …’ recommendations or for promoting bundles of items that are frequently bought together.

1. **Semi-Supervised Learning** – Mixture of unsupervised and supervised learning. The algorithm might first try to create a set of categories and then predict a particular outcome.

In supervised learning, labelled data are used to build a model. However, labelling the

training data for some real-world applications can be difficult and time consuming. In some cases, there are implicit costs associated with obtaining these labels from domain experts. Semi-supervised learning attempts to address this inherent challenge by allowing the model to integrate part or all of the available unlabelled data in its supervised learning.

E.g. identifying people in photos where there are some photos with people already tagged by a user and a large number of untagged photos. A semi-supervised approach uses both the labelled (tagged photos) and unlabelled (untagged photos) data to try and identify people in photos.

1. **Reinforcement Learning** – in this type of machine learning problem we don’t tell the computer what the target of the analysis is, it should be trying to find a course of action that minimises a reward function using trial and error method.

It is the process by which the learner updates the probabilities of taking particular actions on the basis of past rewards received. The learner updates their strategy based on past actions and the associated outcomes.

**Examples** –

* Finding ***strategies for games***. The algorithm can use a representation of the current state of the game as a basis for defining a reward function and look for strategies that improve the chances of winning.
* This process resembles the idea of ***Actuarial Control Cycle*** where adjustments are made periodically based on feedback from past experience to ensure that a particular strategy remains on track. However, with reinforcement learning, it is the machine that updates the strategy based on feedback, without direct human intervention.
* Another popular machine learning technique that involves reinforcement is ***genetic algorithms***, which are based on the idea of selective breeding from biology. These create successive generations of possible solutions. Small random variations (‘mutations’) are introduced into each solution to create the next generation. The solutions are then tested and the ones that perform best are selected to continue to the next generation. After a few generations, a good solution may have been discovered amongst the surviving solutions.

## #Differentiate: Supervised and Unsupervised Learning

1. Paragraphs above
2. The difference between these lies not (as one might think) in the level of involvement of the human researcher in the development of the algorithm, or in the supervision of the machine. Instead, it lies in the extent to which the machine is given an instruction as to the endpoint (or target) of the analysis

## #Stages of Machine Learning-

1. Collect data
2. Explore and prepare data (data cleaning)
3. Scaling
4. Split the data (Train 60%, Validate 20%, Testing 20%)
5. Train the model
6. Validate the outputs (Sensibility of outcome)
7. Evaluate performance (Testing)
8. Improve performance
9. Reproducibility (same output for same inputs every time)

**Cleaning The Data**- replacing missing values, and checking the data for obvious errors is an important stage of any analysis, including machine learning.

## #Discuss the Train-Validation-Test Approach (2019 September)

In machine learning, the convention is to divide the data into two parts. One part is used to train the algorithm and the other part is used to test the output for training.

The train-validation-test approach uses three data sets as follows:

* A training data set which is the sample of data used to fit the model that is, to train the algorithm to choose the most appropriate hypothesis
* A validation data set which is the sample of data used to provide an unbiased evaluation of model fit on the training dataset while adjusting the hyper-parameters these hyper-parameters are often specified in advance and then adjusted/optimised according to the performance of the model on the validation data.
* A test data set which is the sample of data used to provide an unbiased evaluation of the final model fit on the training data set. Under machine learning the results of the modelling exercise are applied to data which was not used to develop the algorithm, so the test data should be representative of the data on which the algorithm is to be used.

A typical split of data is 60% for training, 20% for validation and 20% for testing the principle being that enough data must be selected for the validation and testing sets, with the remainder used for the training set.

**E.g. of classification, regression and clustering problem -**

* An example of a classification problem is a spam filter that classifies emails into the two categories ‘Safe’ or ‘Suspicious’.
* An example of a regression problem is a health awareness app that predicts the user’s life expectancy.
* An example of a clustering problem is a system that groups together postcode areas that tend to have a similar experience of insurance claims.

## #AIC and BIC

* The Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) both impose a penalty for additional parameters. If the sample size is N and the number of parameters is J:
* AIC = deviance + 2\*J BIC = deviance + J\*ln(N)
* Where, the deviance is a measure of the model’s goodness of fit. By minimising the AIC or the BIC, we can achieve a trade-off between obtaining a good fit to the data and minimising the number of parameters in the model.

## #Types Of DataA chart with text and arrows Description automatically generated with medium confidence

* **Attribute (or dichotomous) data** refers to variables whose values consist of just two categories.
* **Ordinal variables** take values that can be ordered in a natural way, whereas the values for **nominal variables** cannot.

## #Bias-Variance Trade-off:

<https://towardsdatascience.com/overfitting-and-underfitting-principles-ea8964d9c45c>

* In supervised learning, the prediction error ‘e’ is composed of the bias, the variance and the irreducible part.
* Bias refers to simplifying assumptions made to learn the target function easily.
* Variance refers to sensitivity of the model to changes in the training data.
* The goal of parameterisation is to achieve low bias (underlying pattern not too simplified) and low variance (not sensitive to specificities of the training data) trade-off.
* It is also important for data analysis to be reproducible and well-documented.

**Reproducibility of research –**

* Data used should be fully described and easily accessible to others
* Any modification to data should be clearly defined with computer codes. Under machine learning, this is called “features engineering”, whereby combination of features are used to create something more meaningful.
* Algorithm selection and model development process should be described with computer codes. This should include model parameters’ selection process and reason of selection.

## #How can we achieve a good balance between bias and variance?

### Regularisation

* A method that can use all the features to choose the final hypothesis g , but will prevent it becoming too complex so that generalisation is poor.
* It is called regularisation or penalisation. This approach exacts a penalty for having too many parameters.
* Let the loss function be L\*(w1, w2, …wJ) . Then the hypothesis ‘g’ will be chosen to be the hypothesis with a set of weights which minimises L\*(w1, w2, …wJ).
* The idea of regularisation, or penalisation, is to add to L\* a cost for model complexity.

### Penalty Function

The more we miscalculate μ, higher until the value of penalty function. Thus, our aim is to minimise the penalty function.

**Why Prefer of Penalty function?**

* The basic MLE can sometimes lead to unrealistic results.
* The estimated values of the parameters can be very sensitive to the sample data and can vary wildly.
* This is most likely to happen when the sample size is small or likelihood function is very flat so that changes in the parameter values make very little difference to the log-likelihood.
* Applying a penalty function encourages the method to produce parameter estimates that are close to the values that would be expected from prior expectations.

**Why lambda\*(μ-a) is used as penalty function?**

* In real life, dataset must contain such values as to suggest that μ is equal to a.
* Thus, by above given penalty function we check the deviation from target values we are expecting.

**Influence of Regularisation Parameter – λ**

* The regularity parameter lambda will be assigned a non-negative value. Otherwise, it would correspond to a reward rather than a penalty.
* If lambda was set to 0, there would be no-penalty and the method would reduce the maximum likelihood estimation.
* If lambda was given a very high value, the penalty dominates the calculations.[Then calculate for lambda tends to infinity and state the value of the estimate].

## #Evaluation Metrics In Machine Learning <https://towardsdatascience.com/evaluation-metrics-18db24a91223>

*Confusion Matrix* A diagram of negative

Description automatically generated

* **Cost of classification –** is a measure of computing cost for classification models. We assign weights to the confusion matrix i.e. we grant *reward* if classification is correct and *penalise* if it is wrong.

A screenshot of a math test

Description automatically generatedLower the cost, better the model.

* **Accuracy –** ratio of correct prediction over total predicted values.

**Choice Between Accuracy And Cost Of Classification –** it depends on the aim of modelling. If the aim is to predict medical disorder, we want few false positives (type1 error) but we can afford to have some false negatives (type2 errors). Thus, we aim for higher accuracy. If the model is related to manufacturing company, we can afford a few errors but the cost is of the essence, thus a model with lower cost.

**Drawback Of Accuracy Is A Measure** - if the data is unbalanced, accuracy will be very high, but its predictive capability of the model may be very bad. Thus, accuracy is not sufficient for this type of classification. It is just not robust enough. To overcome this, we use other metrics. A white rectangular object with black text

Description automatically generatedhere, accuracy is 98% but predictive ability is very poor.

* **True positive rate –** value lies between 0 and 1. The higher the value of TP, the better.
* **False negative rate -** value lies between 0 and 1. The lower the value of TP, the better.
* **True Negative Rate and False Positive Rate** (formula above)
* F1 Score – is the harmonic mean of precision and recall. It eliminates the dilemma of prioritizing precision or recall. It reaches a maximum when precision is equal to recall.

*False Positive = Type 1 Error False Negative = Type 2 Error*

**Akshay Sir CS2B Mock 1 – Take Up Rate**

* Actual take up rate = (TP+FN)/total observation
* Predicted take up rate = (TP+FP)/total observation

A diagram of a diagram

Description automatically generated with medium confidencePrecision –vs.- Recall

**#Thresholding –** When we have a model that is predicting probabilities of classes and not classes themselves, we can set a *threshold value to classify probabilities as 0 or 1.*

### The Kappa Statistic

The kappa statistic adjusts accuracy by accounting for the possibility of a correct prediction by chance alone. This is especially important for datasets with severe class imbalance because a classifier can obtain high accuracy simply by always guessing the most frequent class. The kappa statistic will only reward the classifier if it is correct more often than this simplistic strategy.

0≤ kappa ≤ 1 – indicating perfect agreement between model’s prediction and true values. values less than one indicate imperfect agreement. Depending on how a model is to be used, the interpretation of kappa statistic might vary.

* Poor agreement = less than 0.2
* Fair agreement = 0.2 to 0.4
* Moderate agreement = 0.4 to 0.6
* Good agreement = 0.6 to 0.8
* Very good agreement = 0.8 to 1

, where Pr(a) refers to the proportion of actual agreement and Pr(e) refers to the expected agreement between the classifier and the true values, under the assumption that they were chosen at random.

**Specificity and Sensitivity**

* **Specificity :** is also called true negative rate - measures the proportion of negative examples that were correctly classified.
* **Sensitivity :** is also called true positive rate - measures the proportion of positive examples that were correctly classified.
* Sensitivity and specificity range from zero to one, with values close to one being more desirable. Of course, it is important to find an ***appropriate balance*** between the two—a task that is often quite context-specific.
* Sensitivity and specificity provide tools for thinking about such trade-offs. Typically, changes are made to the model and different models are tested until you find one that meets a desired sensitivity and specificity threshold. Visualizations, such as those discussed later in this chapter, can also assist with understanding the balance between sensitivity and specificity.

**Precision and Recall**

* **Precision** : also known as positive predictive value – is defined as the proportion of positive examples that are truly positive; in other words; when a model predicts the positive class, how often is it correct?

A precise model will only predict the positive class in cases very likely to be positive. It will be very trustworthy. .

* **Recall** : is a measure of how complete the results are – is defined as the number of true positives over the total number of positives.

**Interpretation –** A model with high recall captures a large portion of the positive examples, meaning that it has wide breadth. For e.g. a search engine with high recall returns a large number of documents pertinent to the search query. .

***Prediction mei kitna True ko True bataya is precision – Reality mei se kitna True to True bataya is Recall.***

* Similar to the ***inherent tradeoff*** between sensitivity and specificity, for most real-world problems, it is difficult to build a model with both high precision and high recall. It is easy to be precise if you target only the low-hanging fruit—the easy-to-classify examples. Similarly, it is easy for a model to have high recall by casting a very wide net, meaning that the model is overly aggressive at identifying the positive cases. In contrast, having both high precision and recall at the same time is very challenging. It is therefore important to test a variety of models in order to find the combination of precision and recall that meets the needs of your project.

**The F-measure**

* A measure of model performance that combines precision and recall into a single number is known as the F-measure (also sometimes called the F1 score or the F-score).
* The F-measure combines precision and recall using the harmonic mean, a type of average that is used for rates of change. The harmonic mean is used rather than the more common arithmetic mean since both precision and recall are expressed as proportions between zero and one, which can be interpreted as rates.
* Since the F-measure describes model performance in a single number, it provides a convenient way to compare several models side-by-side. However, this assumes that equal weight should be assigned to precision and recall, an assumption that is not always valid. It is possible to calculate F-scores using different weights for precision and recall, but choosing the weights can be tricky at best and arbitrary at worst. A better practice is to use measures such as the F-score in combination with methods that consider a model's strengths and weaknesses more globally.

## #Receiver Operating Characteristic Curve (ROC)

A graph of a logistic regression

Description automatically generated

* **Trade off** - ROC curve illustrates the trade-off between *recall* and the *false positive rate.*

The receiver operating characteristic (ROC) curve is commonly used to examine the tradeoff between the detection of true positives while avoiding the false positives.

* **History** - ROC curves were developed by engineers in the field of communications. Around the time of World War II, radar and radio operators used ROC curves to measure a receiver's ability to discriminate between true signals and false alarms.
* **Measurement** - The area under the ROC provides a single-figure measure of the efficacy of the model. The further away from the diagonal is the ROC, the greater the area under the curve and the better the model is at correctly classifying the cases.
* **Diagonal line** corresponds to a neutral ‘zero-sum’ test where there is a simple trade-off with any improvement in the true positive rate being matched by an equal deterioration in the false positive rate.
* **Points near the top left** of the graph correspond to a good test where the true positive rate is high and the false positive rate is low.
* **Area of the triangle below****the diagonal** is 0.5 and the area of the whole rectangle is 1 (the maximum possible score for the ROC).

**Uses** - This type of graph is most useful when the test involves a threshold of some kind. E.g. medical test for measuring chemical concentration.

**Area under ROC (AUC)**

* The closer the curve is to the perfect classifier, the better it is at identifying positive values. This can be measured using a statistic known as the area under the ROC curve (AUC).
* The AUC treats the ROC diagram as a two-dimensional square and measures the total area under the ROC curve.
* AUC ranges from 0.5 (for a classifier with no predictive value), to 1.0 (for a perfect classifier).
* Grading System
  + A – Outstanding → 0.9 to 1.0
  + B – excellent/good → 0.8 to 0.9
  + C – acceptable/fair → 0.7 to 0.8
  + D – poor → 0.6 to 0.7
  + E – no discrimination → 0.5 to 0.6
* The AUC value alone is often insufficient to identify a "best" model. In this example, AUC does identify the better model because the ROC curves ***do not intersect***.
* In other cases, the "best" model will depend on how the model will be used.
* When the ROC curves ***do intersect***, it is possible to combine them into even stronger models

<https://towardsdatascience.com/applications-of-different-parts-of-an-roc-curve-b534b1aafb68>

**Estimating Future Performance**

* Some R machine learning packages present confusion matrices and performance measures during the model-building process. The purpose of these statistics is to provide insight about the model's ***Resubstitution Error***, which occurs when the training data is incorrectly predicted in spite of the model being built directly from this data. This information can be used as a rough diagnostic to identify obviously poor performers.
* The resubstitution error is not a very useful marker of future performance, however. For example, a model that used rote memorization to perfectly classify every training instance with zero resubstitution error would be unable to generalize its predictions to data it has never seen before. For this reason, the error rate on the training data can be extremely optimistic about a model's future performance.
* Instead of relying on resubstitution error, a better practice is to evaluate a model's performance on data it has not yet seen.

***K-Fold Cross-Validation***

* Rather than taking repeated random samples that could potentially use the same record more than once, k-fold CV randomly divides the data into ‘k’ completely separate random partitions called ***folds***.
* ***The most common convention is to use 10-fold CV.***

The reason is that empirical evidence suggests that there is little added benefit to using a greater number. For each of the 10 folds (each comprising 10 percent of the total data), a machine learning model is built on the remaining 90 percent of data. The fold's 10 percent sample is then used for model evaluation. After the process of training and evaluating the model has occurred 10 times (with 10 different training/testing combinations), the average performance across all folds is reported.

**Bootstrap Sampling**

* These refer to statistical methods that use random samples of data to estimate properties of a larger set.
* When this principle is applied to machine learning model performance, it implies the creation of several randomly selected training and test datasets, which are then used to estimate performance statistics. The results from the various random datasets are then averaged to obtain a final estimate of future performance.
* ***So, what makes this procedure different from k-fold CV?***

Whereas cross-validation divides the data into separate partitions in which each example can appear only once, the bootstrap allows examples to be selected multiple times through a process of sampling with replacement. This means that from the original dataset of ‘n’ examples, the bootstrap procedure will create one or more new training datasets that also contain ‘n’ examples, some of which are repeated. The corresponding test datasets are then constructed from the set of examples that were not selected for the respective training datasets.

* Using sampling ***with replacement***, as described previously, the probability that any given instance is included in the training dataset is 63.2%. Consequently, the probability of any instance being in the test dataset is 36.8%. In other words, the training data represents only 63.2% of available examples, some of which are repeated. In contrast with 10-fold CV, which uses 90% of examples for training, the bootstrap sample is less representative of the full dataset.
* Because a model trained on only 63.2% of the training data is likely to perform worse than a model trained on a larger training set, the bootstrap's performance estimates may be substantially lower than what will be obtained when the model is later trained on the full dataset. A special case of bootstrapping, known as the 0.632 bootstrap, accounts for this by calculating the final performance measure as a function of performance on both the training data (which is overly optimistic) and the test data (which is overly pessimistic). The final error rate is then estimated as:
* ***One advantage of bootstrap sampling over cross-validation*** is that it tends to work better with very small datasets. Additionally, bootstrap sampling has applications beyond performance measurement.

## #Comment on values of Precision, Recall and F1 score (APRIL’19)

* **Precision values** show that ­­­­­­­\_\_\_\_ (with higher value) is more effective at correctly identifying individuals who do have the disease/feature.
* **Recall values** show that \_\_\_\_ (with higher value) is much better at identifying individuals who do have the disease/feature.
* Whether recall or precision are chosen as measures will depend on whether it is most important to identify all persons who have the disease/feature, or not to unduly worry and treat people who are disease free.
* **As disease/feature is serious** it is best to maximise true positive and minimise false negative. For this we prefer …
* **F1 Score** show that the overall performance of \_\_\_\_\_ (with higher value) is much better than the \_\_\_\_ (with lower value).
  + The F1 score, however, is reasonably robust to the situation where most people do not have the disease, as its calculation does not involve the true negatives.
* **False positive** rates {for both are quite low or clinical>questionnaire} indicating that only a small proportion of individuals who do not have the disease/feature are incorrectly flagged as having it.
* **Patent comment**
  + **As the sample size** is relatively small, the test should be re-performed on a larger population before drawing any conclusions.
  + **Questionnaire/algorithm** is likely to be **easier and cheaper to administrate** and thus may be a **good short-term substitute** until clinical procedure can be widely established.
  + **F1 score** lacks interpretability and hence it should be used in combination with other evaluation metrics.
  + **Performance of well-curated algorithm** also depends on class distribution of target variable, cost of misclassification, and size of training and test sets.

## #K Means Clustering

**Uses of K mean clustering –** recommending products based on similar customers, anomaly detection in customer behaviour.

### What does value of k represent?

‘k’ is a hyper-parameter specifying the number of clusters the algorithm should aim to produce.

**Steps –**

* Step 1 – machine will randomly allocate each data value to a group
* Step 2 – calculate centroid for each cluster
* Step 3 – reassigns if another data value is closer to the centroid until further reassignment cannot be done and distinct categories are created.

**‘The k means clustering algorithm converged after 3 iterations.’ Meaning of convergence?**

The algorithm involves repeatedly finding the centroid of the data points that have been allocated to each cluster and then reallocating the points to the cluster whose centroid they are nearest to. When this process reaches a stage where no further changes are made, the algorithm has converged to the solution.

### Sept 2020 CS2B – Kmeans Clustering Comment

* Graph of data values based on assigned labels shows that the algorithm reasonably identified the two sets of clusters
* Inspection of the table might suggest that the clusters are unlikely to change much.
  + 70 remain in label A and 97 remain in label B
  + 3 move to label A and 30 moves to label B
* Patent Comments –
  + The analyst could potentially improve the results by updating the centres of the clusters, re-calculating the distances, updating the labels and repeating this process until convergence (i.e. until the labels remain constant).
  + The analyst could implement the full kmeans algorithm to ensure convergence of the final clusters.
  + The analyst may want to apply feature scaling / data normalisation to the values of x\_1 and x\_2 so that each of them contributes approximately proportionately to the Euclidean distances and then re-run the analysis.
  + Techniques to measure within-cluster homogeneity could be used to assess the validity of the results.

### Advantage or strength of K means-

* Uses simple principles for identifying clusters which can be explained in non-statistical terms
* Highly flexible and can be adapted to address nearly all its shortcomings with simple adjustments
* Fairly efficient and performs well
* No learning phase as it works with the entire training set.
* Robust to noisy data, no need to filter outliers

### Disadvantage or Weakness of K means-

* Less sophisticated than more recent clustering algorithms
* Not guaranteed to find the optimal set of clusters because it incorporates a random element
* Requires a reasonable guess as to how many clusters naturally exist in the data
* Results are sensitive to units of measurement used
* Clusters may have no natural interpretation
* It can’t be used unless the data have a natural numerical order

### Within-Cluster Sum of Squares by Cluster (faithful dataset in R)

For any particular cluster, this gives the sum of squared distances from each point to the cluster centre. A math equations with numbers and symbols

Description automatically generated where

* Wij: waiting time of ith point in cluster ‘j’
* Eij: eruption time of ith point in cluster ‘j’
* Wj-bar: average waiting time of points in cluster ‘j’
* Ej-bar: average eruption time of points in cluster ‘j’
* nj: number of points in cluster ‘j’

This is the Euclidean distance of ith point from centre in cluster ‘j’.

It is a measure of within-cluster homogeneity.

Smaller the figure, more tightly packed is each cluster.

The ratio between the cluster sum of squares to the total sum of squares is given by

**(between\_SS/ total\_SS = \_\_\_ )**

### Between Cluster Sum Of Squares:

A math equations with numbers and symbols

Description automatically generated with medium confidencewhere, W-bar and E-bar are the overall average of all points and ‘k’ is the number of clusters.

### Total Sum Of Squares:

A math equation with a square and a square

Description automatically generated with medium confidence

The ratio is a measure of how much total sum of squares has been explained by different clusters. Thus, similar to idea of R2 under linear regression.

The closer the ratio is to 1, the greater heterogeneity between clusters and greater the homogeneity within clusters.

Note: We can change the ratio by considering different number of clusters (thus, k is a hyper parameter). Indeed we can force the ratio to equal 1 by simply having each individual point as its own cluster of size 1.

## #Decision Tree

### Greedy splitting

* Creating a binary decision tree is a process of diving up the input space. A ‘greedy’ approach is used to divide the space – called ***recursive binary spitting***.

***Description*** - This is a numerical procedure where all the values are lined up and different split points are tried and tested using a cost function. The split with the lowest cost is selected. All input variables and all possible split points are chosen in a greedy manner i.e. the very best split is chosen each time.

* ***Greedy*** means that at each stage, we just choose the split that appears to be the most effective at separating the remaining elements, without thinking ahead of the consequences this might have on latter splits.
* ***For regression problem*** – we minimise cost function given by sum squared error = Σ(yi – yi-hat)2 i.e. sum:(actual-predicted)2.
* ***For classification problem*** –we minimise Gini Index function given by Gi = Σ pk\*(1-pk)

### Gini Index

The Gini index is a measure of inequality of a distribution that was introduced by the Italian statistician Corrado Gini.

The Gini Index is a measure of the ‘impurity’ of the nodes in the decision trees i.e. the extent to which the final nodes contain a mixture of different data types.

* G = Σ(range of k): pk (1-pk)
* G = 2p1\*p2
* Where, pk is the proportion of training instances with ‘class k’ in the rectangle of interest.
* *For Binary Classification Range***: 0** (perfect class purity) to **0.5** (worst purity)
* *For ‘m’ Categories Range***: 0** (perfect class purity) to **1-(1/m)** (even split of each category, i.e. the proportion of each category in the node is 1/m).
* *LOWER the value of Gini Index, the BETTER.*
* **Process –** when constructing a tree, the Gini score is used at each step to decide how to divide the data. The score is calculated for the candidate splits and when using the greedy approach, the split with the lowest score is chosen. This is the split which gives the *purest overall child nodes*.

*This does not guarantee the best overall classification tree.*

**Identity** 

**Proof** 

**How this identity can be used to calculate a measure of effectiveness of a proposed split point when constructing a decision tree?**

We can measure the effectiveness of a proposed split point by examining the ‘purity’ of the data in each of the child nodes and then calculating an overall measure of the purity of the split.

This can be done by multiplying the proportion of items of type k at each child node by the proportions for each other type j k ≠ and summing. These values are then weighted by the number of items at that node to calculate an overall measure called the Gini index. Using the identity above leads to the following formula:where the sum is taken over all the child nodes of the split and node n is the number of items in each child node.

**Comment on Gini Index value**

* In case of a binary classification problem, the Gini Index can take any value between 0 and 0.5.
* If the points at each node are alike, the value of the index is 0.
* In this case the value of Gini Index is 0.3170.
* **Patent Comment -** So, the algorithm is not very effective at distinguishing between the different types of data points.

### Stopping Criterion

The recursive binary splitting procedure needs to know when to stop splitting as it works its way down the tree with the training data.

Most common – using a ***minimum count*** on the ***number of training members*** (sample size) assigned to each leaf node. If count is less than the minimum, then the split is not accepted and the node is taken as a final lead node. This is another example of a hyper-parameter. This defines how specific to the training data the tree will be. Too specific (e.g. a count of 1) and the tree will overfit the training data and perform poorly with the test data.

### Pruning the tree

* Definition – enhancing the performance of a tree by removing those sections which provide very little power to classify data.
* The *simplest form of pruning* is Reduced Error Pruning: starting at the leaves, each node is replaced with its most popular class. If the prediction accuracy is not affected, then the change is kept.
* The *fastest and simplest pruning method* is to work through each leaf node in the tree and evaluate the effect of removing it using a hold-out test set (another data set that wasn’t used in training the model). Leaf nodes are removed only if it results in a drop in the overall cost function on the entire test. You stop removing nodes when no further improvement can be made.
* *More sophisticated pruning methods* can be used such as cost complexity pruning (also called ‘weakest link pruning’) where a learning parameter (alpha) is used to weigh whether nodes can be removed based on a combination of - the size of the sub-tree and classification performance of resulting reduced tree.

### Advantages of Pruning –

* Easy to interpret
* No overfitting with pruning
* Works for both classification and regression problems
* Can take any type of variables without modifications
* Does not require any data preparation

### Improve Decision tree – (Akshay sir CS2B mock 1)

* We could include left out covariate (input variables) – it could potentially improve the tree. However, we will need to access the impact of any additional input variable. If it improves overall performance, we can include.
* We could produce a range of different tree models using different orders to make the splits and using different split levels for the age variables, to see if there is a different model that performs better.
* In most applications, by aggregating many decision trees, using methods like bagging, random forests, and boosting, the predictive performance of decision trees can be substantially improved.

### Advantage Of Decision Tree –

* Easy to explain
* Closely mirror human-decision making compared to other regression and classification approaches.
* Easy to display graphically
* They can easily handle qualitative predictors without the need to create dummy variables.
* Does not require any pre-processing of data like normalization or standardisation of features – thus it is invariant to data scaling as features are processed separately.
* Decision tree works well with features that are on different scales or a mix of binary and continuous features.
* They are able to generate understandable rules
* They perform classification without requiring much computation.

### Disadvantage Of Decision Tree –

* They have a lower level of predictive ability than other approaches since they aren’t quite robust because a small change in data can cause a large change in the final decision tree.
* Even with use of pre-pruning, they tend to over-fit and provide poor generalisation performance.
* Decision trees are less appropriate for estimation tasks where the goal is to predict the value of a continuous attribute.
* Decision trees are less appropriate for estimation tasks where the goal is to predict the value of a continuous attribute.

## #Naive Bayes

* Naive Bayes is a Supervised Machine Learning algorithm based on the Bayes Theorem that is used to solve classification problems by following a probabilistic approach.
* It is based on the idea that the predictor variables in a Machine Learning model are independent of each other i.e. the outcome of a model depends on a set of independent variables that have nothing to do with each other.
* A Generative model first models the joint distribution, P(x1,x2,…y) and then uses Bayes’ rule to estimate conditional probabilities P(y|x1, x2,…y), which can then be used to predict the category given the covariates.

**Problem** – number of separate probabilities to be computed increases exponentially with the number of covariates.

**Solution** – by assuming the covariates are independent.

A black and white math equation

Description automatically generated**Naive Bayes Classifier**

**Example –** Article Classification using binary word presence, Email spam detection using a similar technique

A table with black text

Description automatically generated

**But why is Naive Bayes called ‘Naive’?**

In real-world problems, predictor variables aren’t always independent of each other; there are always some correlations between them. Since Naive Bayes considers each predictor variable to be independent of any other variable in the model, it is called ‘Naive’.

### Advantage of Naive-Bayes Approach

* + Fast because only probabilities need to be calculated.
  + Easy to apply
  + Requires very little data
  + Good for few category variables
  + If naive assumption work can converge quicker than other models. It can be used for smaller training data.

### Disadvantage of Naive-Bayes Approach

Assumes that the conditional probabilities are independent which can be a poor approximation when the variables are correlated.

**Laplace Estimator**

Example – SMS data

A table with numbers and letters

Description automatically generated

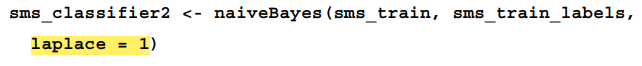
**Problem ?** because P(Spam given groceries) = 0 (zero), the word “groceries” has the ability to effectively nullify and overrule all of the other evidence. The word “groceries” in spam will always veto the other evidence and result in the probability of spam being zero.

**Solution ?** the Laplace estimator adds a small number to each of the counts in the frequency table, which ensures that each feature has a non-zero probability of occurring with each class. ***Typically, the Laplace estimator is set to 1 (one),*** which ensures that each class-feature combination is found in the data at least once.

Note – if you were a devoted Bayesian, you could use a Laplace estimator to reflect a presumed prior probability of how the feature relates to the class.



This is a more plausible result than P(spam) =0 computed when the “groceries” alone determined the result.

**R code** : 

**Using numeric features with Naive Bayes**

Naive Bayes uses frequency tables for learning the data, which means that each feature must be categorical in order to create the combinations of class and feature values comprising the matrix. Since numeric features do not have categories of values, the preceding algorithm does not work directly with numeric data. There are, however, ways that this can be addressed.

One easy and effective solution is to **discretize** numeric features, which simply means that the numbers are put into categories known as **bins**. For this reason, discretization is also sometimes called **binning**. This method works best when there are large amounts of training data.

There are several different ways to discretize a numeric feature. Perhaps the most common is to explore the data for natural categories or **cut points** in the distribution. For example, suppose that you added a feature to the spam dataset that recorded the time of day or night the email was sent, from zero to 24 hours past midnight. Depicted using a histogram, the time data might look something like the following diagram: A graph of a number of bars

Description automatically generated with medium confidence

In the early hours of the morning, message frequency is low. Activity picks up during business hours and tapers off in the evening. This creates four natural bins of activity, as partitioned by the dashed lines. These indicate places where the numeric data could be divided into levels to create a new categorical feature, which could then be used with Naive Bayes.

The choice of four bins was based on the natural distribution of data and a hunch about how the proportion of spam might change throughout the day. We might expect that spammers operate in the late hours of the night, or they may operate during the day, when people are likely to check their email. That said, to capture these trends, we could have just as easily used three bins or twelve.

***If there are no obvious cut points***, one option is to discretize the feature using ***quantiles***. You could divide the data into three bins with tertiles, four bins with quartiles, or five bins with quintiles.

One thing to keep in mind is that discretizing a numeric feature always results in a ***reduction of information***, as the feature's original granularity is reduced to a smaller number of categories. It is important to strike a balance. Too few bins can result in important trends being obscured. Too many bins can result in small counts in the Naive Bayes frequency table, which can increase the algorithm's sensitivity to noisy data.

## Apriori Algorithm – Market Basket Analysis

**Understanding Association Rules** – the result of market basket analysis is a collection of association rules that specify patterns found in the relationships among the items in the itemset. Association rules are always composed from subsets of itemsets and are denoted by relating one itemset on the left-hand side (LHS) of the rule to another itemset on the right-hand side (RHS) of the rule. The LHS is the condition that needs to be met in order to trigger the rule, and the RHS is the expected result of meeting that condition. E.g. {peanut butter, jelly}→ {bread}

**Apriori Algorithm For Association Rule Learning** –

* Given k items that can appear or not appear in a set, there are 2k possible itemsets that could be potential rules. A retailer that sells only 100 different items could have on the order of 2^100 = 1.27e+30 itemsets that an algorithm must evaluate—a seemingly impossible task.
* By ignoring these rare (and perhaps less important) combinations, it is possible to limit the scope of the search for rules to a more manageable size.
* The Apriori algorithm utilizes a simple prior (that is, a priori) belief about the properties of frequent itemsets.
* ***Apriori Property*** → All subsets of a frequent itemset must also be frequent. For example, the set {motor oil, lipstick} can only be frequent if both {motor oil} and {lipstick} occur frequently as well. Consequently, if either motor oil or lipstick is infrequent, then any set containing these items can be excluded from the search.
* The Apriori algorithm uses statistical measures (Support and Confidence) of an itemset's "interestingness" to locate association rules in much larger transaction databases.

**Support and Confidence –**

* Whether or not an association rule is deemed interesting is determined by two statistical measures: support and confidence. By providing minimum thresholds for each of these metrics and applying the Apriori principle, it is easy to drastically limit the number of rules reported.
* ***Support*** → measures how frequently it occurs in the data. For instance, the itemset {get well card, flowers} has the support of 3 / 5 = 0.6 in the hospital gift shop data. Similarly, the support for {get well card} → {flowers} is also 0.6. Support can be calculated for any itemset or even a single item; for instance, the support for {candy bar} is 2/5 = 0.4, since candy bars appear in 40 percent of purchases.

where, N is total number of transactions

* ***Confidence*** → measurement of a rule’s predictive power or accuracy.

Clearly, the order of items X, Y matters a lot because the denominator changes. Essentially, the confidence tells us the proportion of transactions where the presence of item or itemset X results in the presence of item or itemset Y.

*Similarity with Bayesian Probability rules* → In fact, support(A, B) is the same as P(A ∩ B) and confidence(A → B) is the same as P(B | A). It is just the context that differs.

* Rules like {get well card} → {flowers} are known as **Strong Rules** because they have both high support and confidence.

**Basis of association rules** → If {A, B} is frequent, then {A} and {B} must both be frequent. Therefore, if we know that {A} does not meet a desired support threshold, there is no reason to consider {A, B} or any itemset containing {A}; it cannot possibly be frequent.

The process of creating rules then occurs in two phases:

1. Identifying all the itemsets that meet a minimum support threshold.

The first phase occurs in multiple iterations. Each successive iteration involves evaluating the support of a set of increasingly large itemsets. For instance, iteration one involves evaluating the set of 1-item itemsets (1-itemsets), iteration two evaluates the 2-itemsets, and so on. The result of each iteration i is a set of all the i-itemsets that meet the **minimum support threshold**.

1. Creating rules from these itemsets using those meeting a minimum confidence threshold.

Given the set of frequent itemsets, association rules are generated from all possible subsets. For instance, {A, B} would result in candidate rules for {A} → {B} and {B} → {A}. These are evaluated against a **minimum confidence threshold**, and any rule that does not meet the desired confidence level is eliminated.

***-Rest in R script – Book Machine Learning with R 3rd Edition Packt Publishing-***

## #Hyper-Parameters

They cannot be estimated from the data – indeed they must often be defined before an algorithm can be implemented. Hyper-parameters are external to the model and their values cannot be estimated from the data. They are typically specified by the practitioner and may be set using heuristic guidelines. Nevertheless, they are critical to the predictive success of a model.

‘Heuristic’ means that there are no hard and fast rules for these. They are determined using rough guidelines and past experience of what works well, combined with experimentation.

### Differentiate: Parameters and Hyper-parameters (Akshay Sir’s Mock 3)

* Parameters are variables internal to a model. They are either estimated or learned from the data. Their values are used when calculating predictions from the model.

E.g. the linear regression model:

Yi = β0 + β1\*x1i + β2\*x2i+ …+ βj\*xji +ei

It has parameters β0, β1, β2 …βj and σ^2

* Hyper parameters are external to the model. Their values cannot be estimated. Their values often have to be set prior to running an algorithm.

E.g. number of covariates J to include in a regression model, the number of categories in a classification exercise, or the rate at which the model should learn from the data.

### Advantage of large no. of parameters

It can improve the accuracy of the model and predictions, because a model with, more parameters will fit the data more closely than one with fewer parameters.

### Disadvantage of large no. of parameters

There is a risk of over-fitting where the estimates from the model will reflect idiosyncratic characteristics of the “training” data set rather than characteristics which apply to the whole data set. This may lead to the analyst identifying patterns which do not exist. If too many parameters are used the model can become complex and computationally expensive to run. Using too many parameters may lead to model stability issues.

### Hyper-Parameters Relevant To Actuaries

* **Life insurance** – Premium calculation and say we need to define the limit of cigarette consumption to be categorised as heavy smoker.
* **Graduation** - If we are graduating mortality data using a Gompertz-Makeham formula and we need to determine the order of the two polynomials.
* **Time Series** – determining values of p, d ,q for ARIMA process
* **GLM** - If we are applying a generalised linear model, we need to decide on the form of the link function to use.
* **Reinsurance** - If we are modelling large claims in general insurance, we need to specify the cut-off point for a claim to count as ‘large’.
* **Motor insurance** - If we are using geographical area as a rating factor in motor insurance, we need to decide on how many areas to use and which locations these cover.
* **Health** - If we are using a patient’s body mass index (BMI) as a predictor for the outcome of a medical procedure, we may need to specify the dividing lines between weight bands such as underweight (<20), normal (20–25), overweight (25–30) and obese (30+).

## #Generalisation Error

An upper bound can be determined for the magnitude of out-of-sample errors. This shows that, with a large enough training set, the out-of-sample error can be made as small as desired.

## #Weight Regularization

* A common way to mitigate overfitting is to put constraints on the complexity of a network by forcing its weights to take only small values, which makes the distribution of weight values more regular. This is called weight regularization and its done by adding to the loss function of the network a cost associated with having large weights.
* The cost comes in 2 flavours –
  + L1 Regularisation = the cost added is proportional to the absolute value of weight coefficients (the L1 norm of the weights)
  + L2 Regularization – the cost added is proportional to the square of the value of the weight coefficients (the L2 norm of the weights). L2 regularization is also called weight decay in the context of neural networks.
* In Keras, weight regularization is added by passing weight regularizer instances to layers as keyword arguments.

A screen shot of a computer code

Description automatically generated

* ***regularizer\_12(0.001)*** means every coefficient in the weight matrix of the layer will add 0.001\*weight\_coefficient\_value to the total loss of the network.
* ***regularizer\_11(0.001)*** OR ***regularizer\_11\_12(11=0.001, 12=0.001)***

## #Adding Dropout

## Dropout is one of the most effective and most commonly used regularization techniques for neural networks. The dropout rate is the fraction of the features that are zeroed out; its usually set between 0.2 and 0.5. At test time, no units are dropped out, instead, the layer’s output values are scaled down by a factor equal to the dropout rate, to balance for the fact that more units are active than at training time.

A screenshot of a computer

Description automatically generated

In keras, you can introduce dropout in a network via layer\_dropout which is applied to the output of layer right before it : ***layer\_dropout(rate = 0.5)***

## #Choosing a measure of success

|  |  |
| --- | --- |
| Balanced classification problem | * Accuracy * ROC AUC |
| Class-imbalanced problems | * Precision – Recall |
| Ranking or multi-label classification | * Mean average precision |
| Various other type of metrics → | * Mean square error * Correlation – squared correlation * MAD, MAE, RMSE and so on |

## #Choosing the right last-layer activation and loss function

|  |  |  |
| --- | --- | --- |
| Problem type | Last-layer activation | Loss function |
| Binary classification | Sigmoid | Binary cross entropy |
| Multi class single-label classification | Softmax | Categorical cross entropy |
| Multiclass multi-label classification | Sigmoid | Binary cross entropy |
| Regression to arbitrary value | None | MSE |
| Regression to values between 0 and 1 | Sigmoid | MSE or binary cross entropy |

***Note → Regularizing your model and tuning your hyperparameter***

* Add dropout
* Try different architectures – add/remove layers
* Add L1 and/or L2 regularization
* Try different hyperparameters (such as the number of units per layer or the learning rate of the optimizer) to find the optimal configuration
* Optionally, iterate on feature engineering – add/remove features

## #Supervised Learning Techniques -

* + - 1. **Discriminant Analysis –** here, we model conditional probabilities P(y given x1,x2…) directly. The model builds decision boundaries between classes by directly modelling the conditional probabilities P(y given x1, x2…) rather than modelling full joint distribution P(x1,x2… y)
      2. **Binary Logistic Regression** – can be used to predict a binary outcome (e.g. Yes-No). The output here are estimates of conditional probabilities P(Yes given x1, x2 …) and P(No given x1, x2…). This is a discriminant approach as we don’t model joint probabilities P(x1,x2… Yes) and P(x1,x2,…No).

**Note -** Logistic Regression is based on **Logistic Function** which converts an input value (-Inf < x< Inf) to an output value on a continuous scale between 0 and 1.

If we interpret the output as probability, we can convert it to a categorical output by saying that the values exceeding a specified value of probability corresponds to Yes, while smaller values correspond to No.

* + - 1. **Multinomial Logistic Regression** – is an extension to binary logistic regression for outcomes that can take more than two values.
      2. **Perceptrons** and **neural networks** - Use interconnected layers of artificial neurons that can be activated or deactivated in a way that mimics the behaviour of the neurons in animal brains.
      3. **Probit Models -** It (short for ‘probability unit’) produces outputs that can only take one of two values, e.g. Yes / No or 0 / 1.
      4. **Support Vector Machine** - Here, a hyper-plane is selected to separate the points in the input variable space by their class, with the largest margin. The closest data points (defining the margin) are called the support vectors. But the real data cannot be perfectly separated, that is why a ‘C’ defines the amount of violation of the margin allowed. The lower C, the more sensitive SVM is to training data.

**#Advantage of Support Vector Machine -**

* Allows non-linear separation with non-linear Kernels
* Works good in high dimensional space
* Robust to multi-collinearity and overfitting

## #Semi-Supervised Learning

**For example -** a system that aims to *identify pickpockets* operating in a busy shopping street might first identify people who appear several times throughout the day with the same clothing. These people’s faces could then be matched against a database of known offenders.

**Other Examples** – Face detection from images, Target audience classification from tweets

**A semi-supervised learning approach could be carried out as follows:**

(i) Create a first model ‘M1’ using ‘data1’ alone.

(ii) Predict the target values in ‘data2’ using model ‘M1’.

(iii) Create an improved model using the observed target values from ‘data1’ and the predicted target values for ‘data2’ obtained in (ii) as if they were the known target values for the unlabelled dataset.

**Auto encoder** - An auto encoder compresses the raw data by focusing on features that appear to be significant, e.g. it might identify the different types of object that appear in a photo, even though it doesn’t know what they actually are. It can considerably speed up any future modelling analysis.

**#CART algorithm provides a foundation for important algorithms like:**

**Bagged decision trees** - In bagged decision trees, we create random sub-samples of our data with replacement, train a CART model on each sample, and (given new data) calculate the average prediction from each model.

**Random forest** - Random forests apply a method based on averaging a number of randomly generated decision trees.

**Boosted decision trees** - Boosting here refers to a method of repeatedly making small adjustments to improve the effectiveness of a model by reducing the residual error.

## #Parametric -vs- Non-parametric Methods in Machine Learning

**Parametric Models -** In parametric methods, we typically make an assumption with regards to the form of the function ‘f’. for e.g. the unknown function ‘f’ could be linear i.e. of the form f(X) = β0 + β1\*x1 + β2\*x2 + … + βp\*xp. Then we select a model that aligns with this assumption.

Thus, under parametric method in machine learning, we take a model-based approach where we make an assumption with respect to the form of function to be estimated and then we select a suitable model based on this assumption to estimate the set of parameters.

**Disadvantage** – assumption made may not be true. Thus, these methods involves less flexible algorithms and are usually used for less complex problems.

**Advantage** – they are quite fast and require significantly less data than non-parametric models. As they are less flexible and suitable for less complex problems, they are more interpretable.

E.g. Linear Discriminant Analysis, Naive Bayes, Perceptrons.

**Non-Parametric Models -** Non-parametric methods refer to a set of algorithms that do not make any underlying assumptions with respect to the form of the function to be estimated.

**Advantage** - They tend to be more accurate as they seek to best fit the data points. They are quite flexible and can lead to better model performance since no assumptions are being made about the underlying function.

**Disadvantage** - But they require a very large number of observations. Additionally, these methods tend to be less efficient when it comes to training the models. Furthermore, they may sometimes introduce overfitting. These algorithms tend to be more flexible, they may sometimes learn the errors and noise in a way that they cannot generalise well to new, unseen data points.

E.g. Support Vector Machines, K-Nearest Neighbours.

# Missing Data On Machine Learning

## Reasons For Missing Values

* Human error when processing data
* Machine error due to malfunctioning of equipment
* Respondents refusal to answer certain questions
* Drop out in studies
* Merging unrelated data

## Issues Caused By Missing Values

* Performance degradation
* Data analysis problems
* Biased outcomes
* Seriousness of issue depends on the volume of missing data, pattern and the underlying mechanism in the missingness of data

## Missing Data Patterns And Mechanisms

Missing data patterns describe which values are missing and observed in a data set. The 3 patterns that appear most are Univariate, Monotone and non-Monotone.

A diagram of different colors

Description automatically generated with medium confidence

* **Univariate –** missing data is univariate when there is only one variable with missing data. It arises most in experimental studies.
* **Monotone –** missing data is said to be monotone if the variables in the data can be arranged, the pattern is usually associated with a longitudinal studies where members drop out and never return. It is easier to deal with as these are easily observable.
* **Non Monotone –** missing data whereby the missingness of one variable does not affect the missingness of any other variables.

**Missing Data Mechanisms**

A close up of a text

Description automatically generated

* **Missing Completely at Random (MCAR) :** when missing observations are not reliant on the observed and unobserved measurements.
* **Missing at Random (MAR) :** the likelihood of a missing value in MAR is only related to the observable data. Mostly encountered in health science studies data sets. Under this mechanism, missing values can be handled by observed predictor variables.
* **Missing not at Random (MNAR) :** here missing data equally depends on missing and observed values. In this method, handling the missing values is usually impossible, as it depends on the unseen data. This mechanism is mostly applied in different domains of biomedicine, psychology and educational data sets.

**A close up of text

Description automatically generated**

## Missing Value Approaches

### Delete – Discard – Remove

**#Imputation :** replacing missing values by some predicted values

### Simple imputation

Replace with mean, median, mode of available data. It produces bias or unrealistic results on a high-dimensional data set.

### Regression imputation

Build a regression model using observed variables and predict the missing values.

### Hot-deck imputation

Handles missing values by matching the missing values with other values in the data set on several other key variables have complete values.

**1st method** – randomly select a pool of all cases. This pool is called the donor pool, that is identical to the cases with missing data on many variables and chooses one case randomly out of that pool. The missing value is then replaced by data from the randomly chosen cases.

**2nd method** – involves replacing the closest donor neighbour rather than selecting one donor from a pool of donors. This method disregards the variability in missing data.

**3rd method** – ***Weighted Random Hot-deck***. This method does not limit the number of times a donor is nominated; however, the donors are chosen randomly from the donor pool.

**4th method –** ***Weighted Sequential Hot-deck***. This method puts a restriction on the amount of time a donor can be chosen to prevent the same donor to be paired with a large quantity of recipients.

**Advantages of Hot-Deck Method**

* Very popular as it results in a rectangular data – avoids cross-user inconsistency
* Does not depend on model fitting for missing value to be replaced – making it less delicate to model specification as compared to a method built on a parametric model, for instance regression model.
* It also decreases bias in non-response.

### Expectation-Maximization

This is an iterative method for handling missing values in numerical datasets, the algorithm uses an “*impute, estimate and iterate until convergence”* approach. Every iteration includes 2 stages – expectation and maximisation.

Expectation estimates missing values given observed data, whereas in Maximisation, the present estimated values are used to maximize the probability of all data.

This method results being specific to idiosyncrasies in the dataset and in sampling or are reflective oh hypothetical expectations i.e. overfitting. In high dimensional dataset, it lead to expensive matrix computations.

In general, single imputation methods are mostly biased and do not represent the vulnerability associated with the missing values.

### Multiple Imputation

Here the distribution of observed data is utilized to approximate numerous values that reflect uncertainty around the true value, and this method was mostly implemented to solve the limitations of single imputation.

**Method :** The analysis is done on a data set using various missing data techniques, and the average parameter estimates across ‘M’ samples is computed into a single point estimate.

**3 distinct phases –**

1. missing data is handled in M resulting samples in M complete data sets.
2. The M complete data sets are then analysed
3. The results of all the M imputed data sets are combined for final imputation result.

**Drawbacks**

* Performance may be negatively affected when carrying out imputation on real data such as survey data, clinical data and industrial data which may be characterised by a high rate of missingness and a great number of factors that are not necessarily linearly related.
* Traditional multiple imputation methods seem to perform poorly on high dimensional data.
* Caution should be made for continuous-based techniques when imputing categorical-data as this may lead to biased results.

**Some multiple imputation approaches**

* **Least Square SVM**
* **Clustered Z-score Least Square SVM** (classification problems)
* Methods gave unbiased estimates and standard errors

### Machine Learning Inspired Imputation

This involves developing a predictive approach to handle missing values using supervised or unsupervised learning.

* **K Nearest Neighbour Classification (KNN)**

The KNN algorithm works by classifying the nearest neighbours of missing values and use those neighbours for imputation using a distance measure between instances. E.g. of distance instances – Minkowski distance, Manhattan distance, Cosine distance, Jaccard distance, Hamming distance and Euclidean distance (most effective and productive – widely used).

Advantage – flexible in both discrete and continuous data. Outperforms all other techniques discussed so far.

Disadvantage – low precision, introduces false associations where they do not exist, increases computational time. There is no proven common resolution to select the optimized KNN parameters. It often neglects the significance of influences of the missingness mechanisms.

Related techniques – Iterative KNN imputation (take advantage of correlation of attributes by using grey relational grade instead of Euclidean distance), Incomplete-instance based imputation approach called CVBkNN (utilise cross validation to improve parameters for each missing value), EvlKNNImpute.

Another study incorporated correlation matrix for KNN algorithm design. The lease-square loss function was used to minimize the reconstructed error and reconstruct every test data point by using all training data points. This study did not consider the influence of missingness mechanisms and patterns on imputation performance.

* Support vector machine (SVM)

The SVM, for a labelled training sample, efforts to discover an optimal separating hyper-plane such that the distance from the hyper-plane to the nearest data points is maximized.

* Decision tree

Decision trees can produce a complex tree that tend to be time consuming but have a low bias. A decision tree and forest technique for the imputation of categorical and numerical missing values was proposed. The technique identified horizontal segments in the data set where the records belonging to a certain segment had higher similarity and attribute correlations. The missing data were then imputed using the similarity and correlations. The three most used decision tree learning algorithms are: ***ID3, C4.5 and CART***.

A text on a page

Description automatically generatedA close up of a text

Description automatically generated

A text on a page

Description automatically generated

* Clustering imputation
* Ensemble Methods

**Performance metrics**

* Mean Absolute Error (MAE) A mathematical equation with numbers and symbols

  Description automatically generated Measures average difference between imputed and true values.
* Mean Squared Error (MSE) A mathematical equation with numbers and symbols

  Description automatically generated Measures the average squared difference between the predicted missing values and actual values.
* Root Mean Squared Error (RMSE) A math symbols with numbers and symbols

  Description automatically generated with medium confidence Measures the standard deviation of the differences between estimated missing values and observed values.

Note: ‘m’ is the number of observations.

* Area Under the Curve (AUC) A math equation with black text

  Description automatically generated AUC is the representation of the degree or measure of separability and is used as a summary of the Root Receiver Operator Characteristic (ROC) curve. The AUC is represented by the true positive rate (TPR) and the false positive rate (FPR).